# Program size complexity of self-assembled rectangles: constant and variable temperature *

Eric Goles, Ivan Rapaport†and Cristobal Rojas
Departamento de Ingeniería Matemática and Centro de Modelamiento Matemático
UMR 2071-CNRS, Universidad de Chile, Santiago, Chile

**Abstract**

In this paper we consider the Tile Assembly Model proposed by Rothemund and Winfree. Let $\lambda_n$ be such that ${\lambda_n}^{\lambda_n} = n$. We call "thin" rectangles those having dimensions $k \times n$ with $1 \leq k \leq \lambda_n$. We prove that, when the temperature is constant, the program size complexity of these rectangles is $\Theta(n^{1/k})$. On the other hand, if we allow a single change in the temperature, the complexity decreases dramatically to $O(\lambda_n)$.

## 1 Introduction

In this paper we consider the Tile Assembly Model proposed by Rothemund and Winfree [6, 8]. The individual components are modelled as square tiles. These tiles "float" on the two dimensional plane. They cannot be rotated. Each side of a tile has a specific "glue". When two tiles collide they stick if their abbuting sides have compatible glues.

The main difference between this model and the classical one [7] is that here we take into account a global parameter: the temperature $\mathcal{T}$. More precisely, two tiles can stick as far as the strength of the glue in their abbuting sides is high enough with respect to $\mathcal{T}$.

The problem tackled by many authors is to construct a set of tiles that self-assemble into a particular shape. The program size complexity of a given shape $S$ is the minimal number of distinct tiles required to produce $S$. Rothemund and Winfree [6] proved that the program size complexity of an $n \times n$ square is $\Theta(\log(n)/(\log\log(n))$. Adleman *et al.* [1] showed that the problem of finding a minimal tile system that uniquely produces a given shape is NP-hard in general but, for some families of shapes, it can be solved in polynomial time. They developed algorithms for trees, squares, "thick rectangles" (rectangles where the width is at least logarithmic in the height), and they ask for "thin rectangles". In another work, Adleman *et al.* [2] studied infinite ribbons and undecidability results were obtained. 3-dimensional structures have also been "constructed" by self-assembly [3, 4].

*Partially supported by projects FONDAP on Applied Mathematics and FONDECYT 1020611 (I.R.)
†Corresponding author: irapapor@dim.uchile.cl

In this paper we consider $k \times n$-rectangles. Our motivation is twofolds. On one hand we analyze how does the complexity vary with $k$. On the other hand we show how to significatively decrease the complexity by the following simple protocol:

- Construct a $(k + \lambda_n) \times n$-rectangle with temperature 2.
- Increase the temperature to 4.
- The maximal 4-stable sub-shape of the $(k + \lambda_n) \times n$ rectangle is a $k \times n$ rectangle. This is the figure that will "tolerate" the higher temperature. The remaining part will disintegrate.

We would like to point out that this model has been conceived not only as a way to form particular shapes, but also as a way to execute computations [5, 9, 10].

# 2 Notation and definitions

A tile $t$ is an oriented square with glues in its north, south, east and west edges. Formally, $t = (\sigma_N, \sigma_E, \sigma_S, \sigma_O) \in \Sigma^4$. Each type of glue $\sigma \in \Sigma$ has a strength $g(\sigma) \in \mathbb{N}$. Tiles are respresented as in the next figure. The number of lines in front of the color corresponds to the strength of it. There is one exception to that convention: no lines mean strength 1.

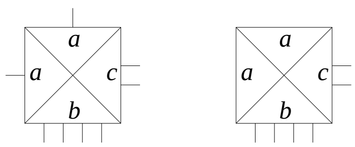


**Figure 2.1: Two ways of representing the same tile. One line or no line correspond to the same strength of the color (strength 1).**

Let us consider a set of tiles $T$, a particular tile $s$ (called the seed), a temperature $\mathcal{T}$, and a strength function $g$. This 4-tuple $(T, s, g, \mathcal{T})$ is called a tile system. A configuration $C$ is an assignment of tiles to a connected region of the plane so that adjacent tiles have the same glue in their abbuting edges. A configuration $C$ is $\mathcal{T}$-stable if it can not be partitioned into two subconfigurations $C_1$ and $C_2$ such that the interaction strength $g(C_1, C_2) < \mathcal{T}$. The value $g(C_1, C_2)$ is the sum of all the strength of the sides of the bound between $C_1$ and $C_2$ (see Figure 2.2).

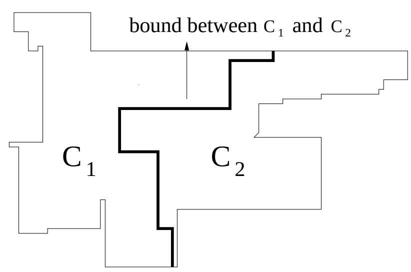


**Fig. 2.2: Bound between $C_1$ and $C_2$.**

The dynamics is as follows. The initial configuration is the seed $s \in T$ placed at the origin of the plane. We denote this configuration by $s$. Let $C$ be a $\mathcal{T}$-table configuration. We write $C \rightarrow C'$ if $C'$ is obtained from $C$ by adding a single tile such that $C'$ is $\mathcal{T}$-table. We say that the configuration $C$ is produced by the tile system $\mathbf{T}$=$(T, s, g, \mathcal{T})$ if $s \rightarrow^* C$.

A tile system $\mathbf{T}$ will be called deterministic if for every cell $(i, j)$ of the plane and for every two configurations $C', C''$ produced by $\mathbf{T}$ covering $(i, j)$, we have $C'(i, j) = C''(i, j)$.

If we view the dynamics of a tile system as a directed graph with the nodes being the $\mathcal{T}$-stable configurations produced by $\mathbf{T}$ linked according to $\rightarrow$, then it is not difficult to notice that the graph is a lattice if and only if the dynamics is deterministic. It is interesting to point out that if we just increase the temperature of the tile system then the new dynamics corresponds to a sublattice of the original one.

We are interested here in deterministic dynamics. Moreover, we focus our attention on finite lattices. These have a global maximum which corresponds to the final configuration. In this case we say that this final configuration is uniquely produced. Recall that the shape of this configuration is the object we are interested in.

A $k \times n$-rectangle is the rectangle that has $k$ columns and $n$ rows. The program size complexity of a $k \times n$-rectangle will be denoted by $C_{\mathcal{T}}(k, n)$ and corresponds to the minimal number of tiles that uniquely produce the rectangle with temperature $\mathcal{T}$.

## 3 Constant temperature and "thin" rectangles

We choose, as Rothemund and Winfree did [6], temperature $\mathcal{T}$= 2. Notice that with temperature 1 the model becomes useless. In fact, in that case, the complexity of every shape becomes the number of its cells (since otherwise the object would keep indefinitely growing). For sake of clarity we will explain in detail the case of the $3 \times n$ rectangle. Extensions to the $k \times n$-rectangle, with $k << n$, are rather straightforward.

**Proposition 3.1.** $C_2(3, n) \geq n^{1/3}$.

*Proof*: Let $T$ be the set of tiles of the tile system that uniquely produces the rectangle. If $|T| < n^{1/3}$ then there will be two repeated rows. This would imply that the object will grow indefinitely. This is a contradiction. $\square$

**Proposition 3.2.** $C_2(3, n) = O(n^{1/3})$.

*Proof*: We will show a tile system $T_{(3,n)}$ that uniquely produces the $3 \times n$-rectangle. The main idea behind the proof is a generalization of the "counting principle" used by Rothemund and Winfree [6]. The tile system simulates a binary counter in which the $i^{th}$ row above the origin (seed row) represents the binary integer $i$. The limitation is that the total number of distinct integers that can be represented -what determines the maximum height of the rectangle ($2^3$ in this case)- depends on the width. To solve this, we use a base $b$ such that the height $n$ written in $b$-ary can be represented in terms of the width (in this case we need $b = \ulcorner n^{1/3} \urcorner$ and the tile system counts in base $b$ representing in each row an integer $i$ between 0 and $n - 1$).

We will show the construction of a $3 \times n$-rectangle using $4b+2$ tiles. Each tile has, in addition to its four labels, a "main label" that gives the semantic information explaining its "function".

The tile system works as follows.

The seed row encodes the number $b^3 - n$ and is formed by 2 special tiles added to the seed tile. The row $i+1$ assembles to the row $i$ increasing the counter by 1.

The other $4b$ tiles appear in the next figure. They are classified into 4 types: Type 1, Type 2, Type 3, Rightmost. In addtion, by looking horizontally, another classification arises: 0-tiles, 1-tiles, . . . , (b-1)-tiles.

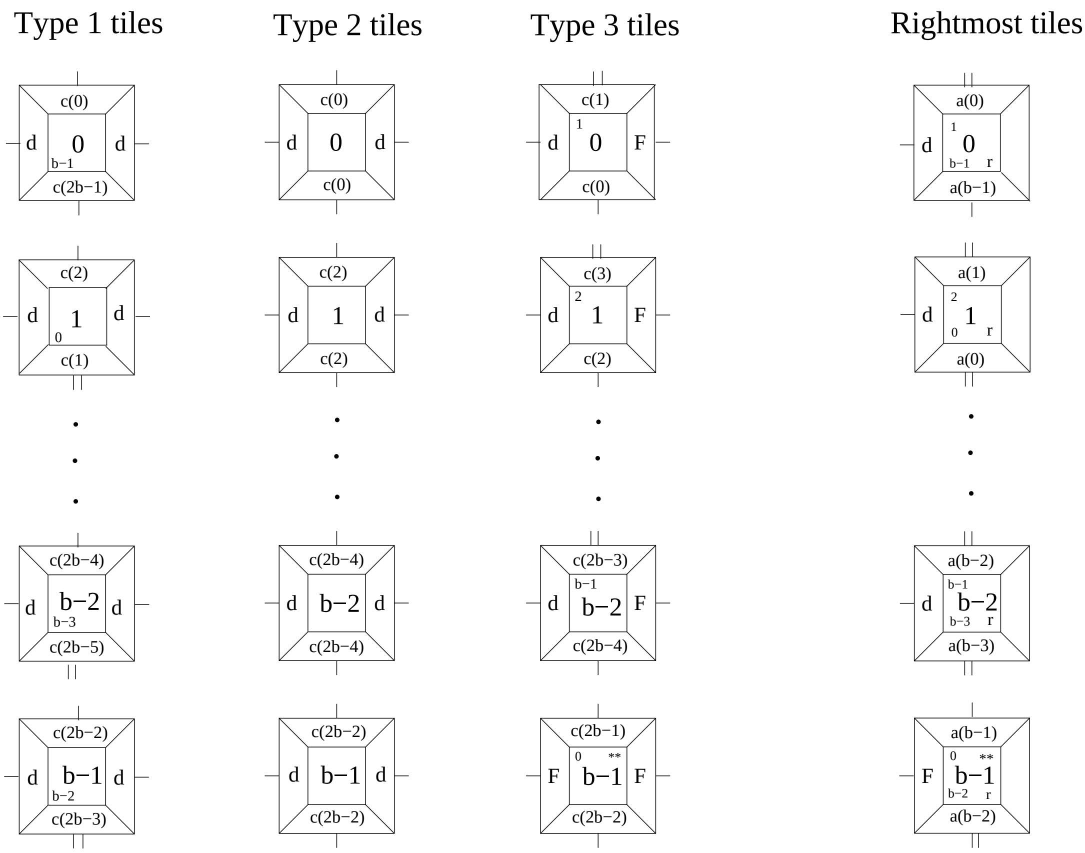


**Figure 3.2: Set of tiles that works as a counter. These tiles are used to construct a $3 \times n$-rectangle.**

In the next figure it is explained how the semantic information of the "main label" must be understood.

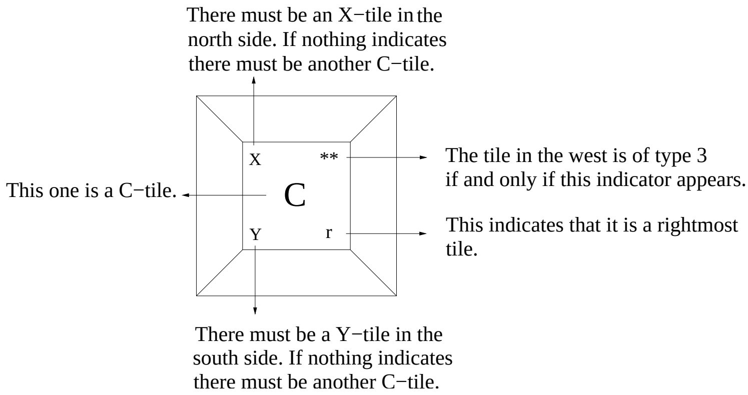


**Fig 3.3: A $C$-tile and its semantic information.**

Since the north side strength of the $(b-1)$-tiles is 1, when the integer $n-1$ is encoded, the counter stops. By following the semantic rules it is not difficult to check that the desired rectangle is assembled. □

Figure 3.4 shows the case $n = 25$.

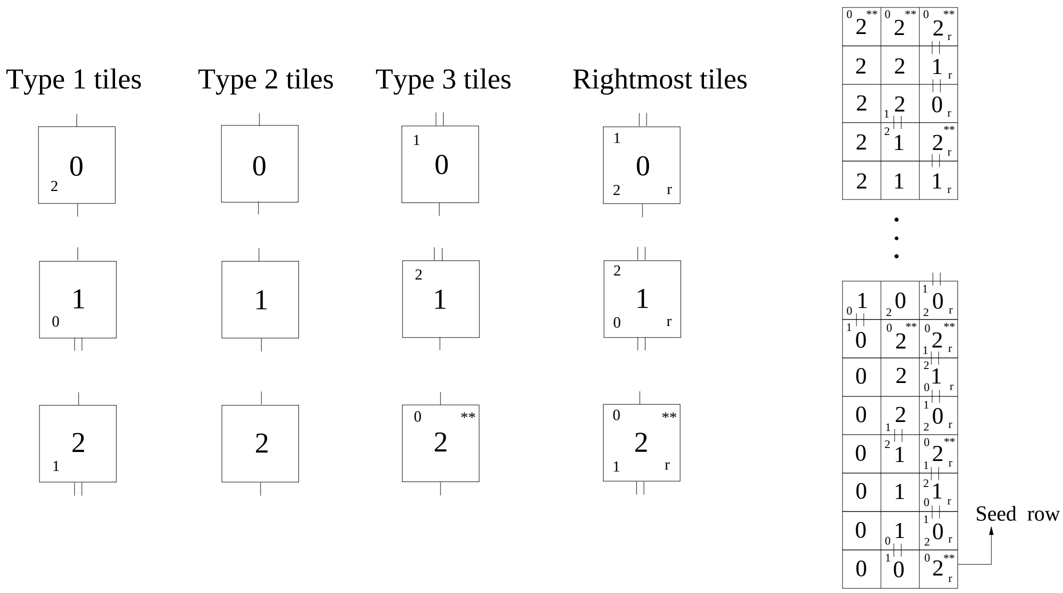


**Fig 3.4: Set of tiles that uniquely produces a $3 \times 25$-rectangle. The east and west sides have all strength 1. In this case $b = 3$. The seed row encodes the number $3^3 - 25 = 2$ or $0\ 0\ 2$ if we write it in $b$-ary.**

**Corolary 3.1.** $C_2(3, n) = \Theta(n^{1/3})$

# 4 Constant temperature and "thick" rectangles

Let us consider the value $\lambda_n$ such that ${\lambda_n}^{\lambda_n} = n$. In this Section we will show that, for a width $k \geq \lambda_n$, the program size complexity $C_2(k,n) = O(\ulcorner \lambda_n \urcorner)$. This upper bound does not depend on $k$.

**Proposition 4.1.** $C_2(\lambda_n, n) = \Theta(\lambda_n)$.

*Proof*: The construction of a $\lambda_n \times n$-rectangle is straightforward from the construction shown in Section 3. The set of tiles shown in Figure 3.2 also works in this general case. We just need to put $b = \ulcorner n^{1/\lambda_n} \urcorner = \ulcorner \lambda_n \urcorner$ and add $\ulcorner \lambda_n \urcorner - 1 \leq b$ special tiles to form the seed row from the seed tile. Therefore $4b + \lambda_n - 1 \leq 5b$ tiles are used. $\square$

It is important to observe that, in order to construct the $k \times n$-rectangle, we needed on one hand the "counter" set of $O(\ulcorner n^{1/k} \urcorner)$ distinct tiles and, on the other hand, $k-1$ special tiles to form the seed row. In other words, for having a complexity $O(n^{1/k})$, we needed the value of $k$ to be at most $n^{1/k}$. This explains the appearance of the critical value $\lambda_n$.

**Proposition 4.2.** $C_2(\lambda_n + k, n) = O(\lambda_n)$.

*Proof*: First we need $2C_2(\lambda_n, n)$ tiles in order to construct a particular $\lambda_n \times n$ rectangle. The particularity lies in the fact that the rightmost column is divided into two kinds of tiles: the $k$ upper ones and the $n-k$ lower ones. The idea is rather simple: the seed row encodes in its north side the number $b^b - k$ and the number $b^b - (n-k)$ in its south side (see Figure 4.1).

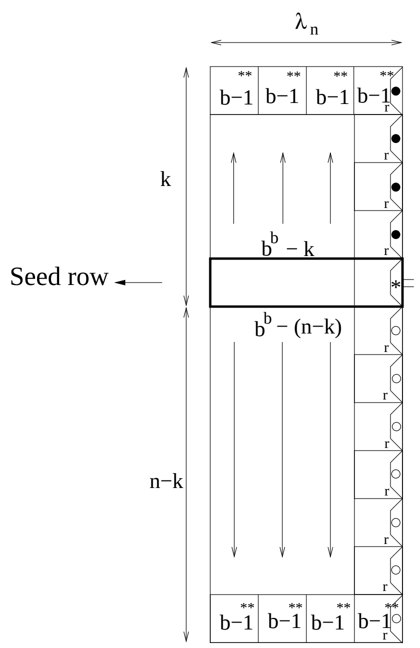


**Figure 4.1 Two counters are used to construct a $\lambda_n \times n$-rectangle.**

We need another 4 "filling" tiles. Their job is to complete a $k \times n$ rectangle to the right of the special column. The particular tile of the column -the one with strength 2 in its right side- is a key signal for the "filling" process (see Figure 4.2). $\square$

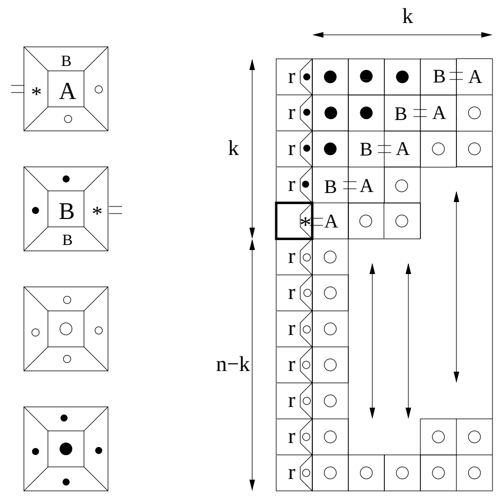


**Figure 4.2: The 4 "filling" tiles and the way they act starting from the particular column.**

# 5 Increasing the temperature to $\mathcal{T} = 4$

Suppose that we have a $\mathcal{T}$-stable configuration of a given tile system. And suppose that we decide to increase the temperature to $\mathcal{T} + \Delta$. Without loss of generality we asume the seed tile to be fixed at the origin (or, equivalently, the origin *is* the place where the seed is located).

A new dynamics will emerge. If $\Delta$ is high enough then the configuration will lose its stability and will start to break. The following Proposition says that it doesn't matter how this process goes on. The output, which is for us the configuration that contains the seed, will always be the same.

**Proposition 5.1.** *Let $D$ be a $\mathcal{T}$-stable configuration produced by a tile system $\boldsymbol{T} = (T, s, g, \mathcal{T})$. Then there exists a unique subconfiguration $M \subseteq D$, $\mathcal{T} + \Delta$-stable maximal (w.r.t. $\subseteq$), that is produced by $\boldsymbol{T}$.*

*Proof*: The seed is a $\mathcal{T} + \Delta$-stable configuration. Let $C_1$ and $C_2$ be two $\mathcal{T} + \Delta$-stable configurations produced by the tile system. The determinism allows us to consider the superposition of $C_1$ and $C_2$ as a feasible configuration. Moreover, this configuration is $\mathcal{T} + \Delta$-stable. $\square$

In this new context we can define the program size complexity of a $k \times n$-rectangle, denoted by $C_{\mathcal{T}}^{\mathcal{T}+\Delta}(k, n)$, to be the minimal number of distinct tiles that uniquely produce a shape whose maximal $\mathcal{T} + \Delta$-stable subconfiguration is the $k \times n$-rectangle.

Let us return to our original problem in which we wanted to construct a $3 \times n$-rectangle. With constant temperature $\mathcal{T} = 2$ we needed $\Theta(n^{1/3})$ distinct tiles.

If instead we construct a $(\lambda_n + 3) \times n$-rectangle whose maximal 4-stable subconfiguration is a $3 \times n$-rectangle and later we increase the temperature to $\mathcal{T} = 4$, we will need only $O(\lambda_n)$ tiles and the final goal will remain unchanged. In other words, by playing with the temperature, the program size complexity of the rectangle decreases dramatically. For instance, if $n = 16.000.000$, the value in the constant approach is 252 while in the variable one is 8.

**Proposition 5.2.** $C_2^4(3, n) = O(\lambda_n)$

*Proof*: In the Figure 5.1 it is shown how to construct a $(\lambda_n + 3) \times n$-rectangle whose maximal 4-stable subconfiguration is a $3 \times n$-rectangle. A $\lambda_n \times n$-rectangle is assembled using the previous approach. Special tiles enlarge the seed row to construct (with stronger tiles) a $3 \times n$-rectangle. □

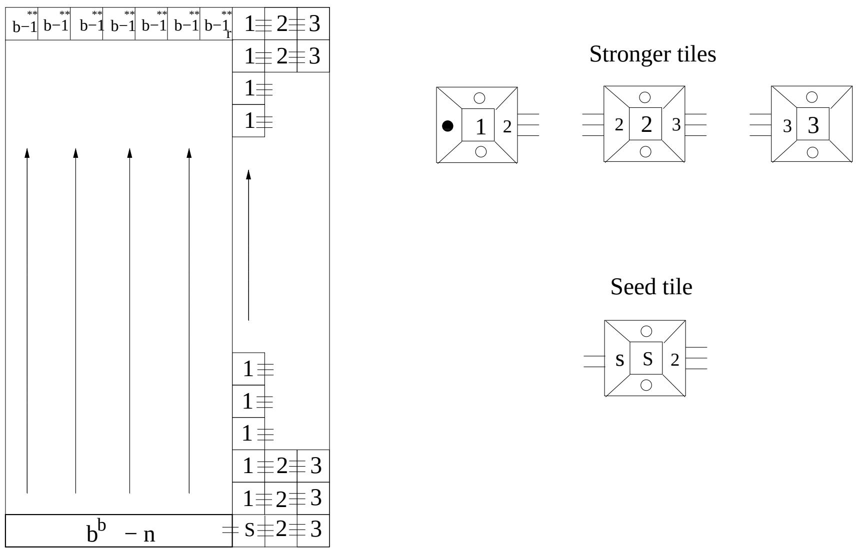


**Figure 5.1: The stronger tiles assemble into a $3 \times n$-rectangle by sticking to the right of the $\lambda_n \times n$-rectangle. Notice that the seed belongs to the $3 \times n$-rectangle.**

# 6 Conclusions

The value $\lambda_n$ is such that $\lambda_n^{\lambda_n} = n$. Our contribution is the study of the program size complexity of the $k \times n$-rectangles with $1 \leq k \leq \lambda_n$. Our results can be stated as follows:

$C_2(k, n) = \theta(n^{1/k})$

and

$C_2^4(k, n) = O(\lambda_n)$.

Tighter bounds should be obtained in the future.

## References


[1] L. Adleman, Q. Cheng, A. Goel, M. Huang, D. Kempe, P. Moisset de Espanés and P. Rothemund, "Combinatorial optimization problems in self-assembly", Proc. 34th ACM Symposium on Theory of Computing (STOC), 2002, 23-32.

[2] L. Adleman, J. Kari, L. Kari and D. Reishus "On the decidability of self-assembly of infinite ribbons", Proc. 43rd IEEE Symposium on Foundations of Computer Science (FOCS), 2002, 530-537.

[3] N. Jonoska, S. Karl and M. Saito, "Graph structures in DNA computing". In *Computing with bio-molecules, theory and experiments*, (editor Gh. Pauen), Springer-Verlag (1998), 93-110.

[4] M-Y Kao and V. Ramachandran, "DNA self-assembly for constructing 3D boxes", Algorithms and Computation 12th International Symposium (ISAAC), 2001, Lecture Notes in Computer Science 2223, 429-440.

[5] C. Mao, T. LaBean, J. Reif and N. Seeman, "Logical computation using algorithmic self-assembly of DNA triple-crossover molecules", Nature 407, 493-496, (2000).

[6] P. Rothemund and E. Winfree, "The program size complextity of self-assembled squares", Proc. 33rd ACM Symposium on Theory of Computing (STOC), 2001, 459-468.

[7] H. Wang, "Proving theorems by pattern recognition II", Bell Systems Technical Journal, 40 (1961), 1-42.

[8] E. Winfree, "Algorithmic self-assembly of DNA", PhD thesis, California Institute of Technology, Pasadena, (1998).

[9] E. Winfree, T. Eng and Grzegorz Rozenberg, "String tile models for DNA computing by self-assembly", DNA Computing 2000, 63-88.

[10] E. Winfree, X. Yang and N. Seeman, "Universal computation via self-assembly of DNA: some theory and experiments", Proceedings of the Second Annual Meeting on DNA Based Computers, 1996, 172-190.